

**Министерство образования и науки Российской Федерации**  
федеральное государственное автономное образовательное учреждение  
высшего образования  
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

---

Институт Энергетический  
Направление подготовки 13.04.02 «Электроэнергетика и электротехника»  
Кафедра Электрические сети и электротехника

**МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ**

| Тема работы                                                          |
|----------------------------------------------------------------------|
| Создание программы дорасчета параметров ТИ и ТС (Калькулятор ТИ, ТС) |
| УДК 004.42:621.397.13.08.001.24                                      |

Студент

| Группа | ФИО                     | Подпись | Дата |
|--------|-------------------------|---------|------|
| 5AM5Д  | Руденко Николай Юрьевич |         |      |

Руководитель

| Должность             | ФИО           | Ученая степень,<br>звание | Подпись | Дата |
|-----------------------|---------------|---------------------------|---------|------|
| Старший преподаватель | Васильев А.С. | к.т.н.                    |         |      |

**КОНСУЛЬТАНТЫ:**

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

| Должность | ФИО           | Ученая степень,<br>звание | Подпись | Дата |
|-----------|---------------|---------------------------|---------|------|
| доцент    | Фигурко А. А. | к.э.н.                    |         |      |

По разделу «Социальная ответственность»

| Должность | ФИО          | Ученая степень,<br>звание | Подпись | Дата |
|-----------|--------------|---------------------------|---------|------|
| Ассистент | Невский Е.С. |                           |         |      |

**ДОПУСТИТЬ К ЗАЩИТЕ:**

| Зав. кафедрой | ФИО           | Ученая степень,<br>звание | Подпись | Дата |
|---------------|---------------|---------------------------|---------|------|
| ЭСиЭ          | Прохоров А.В. | к.т.н.                    |         |      |

Томск – 2017 г.

## ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ООП

| Код<br>результата                   | Результат обучения<br>(выпускник должен быть готов)                                                                                                                                                                                                                       |
|-------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Профессиональные компетенции</b> |                                                                                                                                                                                                                                                                           |
| <b>Р1</b>                           | Применять соответствующие гуманитарные, социально-экономические, математические, естественно-научные и инженерные знания, компьютерные технологии для решения задач расчета и анализа электрических устройств, объектов и систем.                                         |
| <b>Р2</b>                           | Уметь формулировать задачи в области электроэнергетики и электротехники, анализировать и решать их с использованием всех требуемых и доступных ресурсов.                                                                                                                  |
| <b>Р3</b>                           | Уметь проектировать электроэнергетические и электротехнические системы и их компоненты.                                                                                                                                                                                   |
| <b>Р4</b>                           | Уметь планировать и проводить необходимые экспериментальные исследования, связанные с определением параметров, характеристик и состояния электрооборудования, объектов и систем электроэнергетики и электротехники, интерпретировать данные и делать выводы.              |
| <b>Р5</b>                           | Применять современные методы и инструменты практической инженерной деятельности при решении задач в области электроэнергетики и электротехники.                                                                                                                           |
| <b>Р6</b>                           | Иметь практические знания принципов и технологий электроэнергетической и электротехнической отраслей, связанных с особенностью проблем, объектов и видов профессиональной деятельности профиля подготовки на предприятиях и в организациях – потенциальных работодателях. |
| <b>Универсальные компетенции</b>    |                                                                                                                                                                                                                                                                           |
| <b>Р7</b>                           | Использовать знания в области менеджмента для управления комплексной инженерной деятельностью в области электроэнергетики и электротехники                                                                                                                                |
| <b>Р8</b>                           | Использовать навыки устной, письменной речи, в том числе на иностранном языке, компьютерные технологии для коммуникации, презентации, составления отчетов и обмена технической информацией в областях электроэнергетики и электротехники.                                 |
| <b>Р9</b>                           | Эффективно работать индивидуально и в качестве члена или лидера команды, в том числе междисциплинарной, в области электроэнергетики и электротехники.                                                                                                                     |
| <b>Р10</b>                          | Проявлять личную ответственность и приверженность нормам профессиональной этики и нормам ведения комплексной инженерной деятельности.                                                                                                                                     |
| <b>Р11</b>                          | Осуществлять комплексную инженерную деятельность в области электроэнергетики и электротехники с учетом правовых и культурных аспектов, вопросов охраны здоровья и безопасности жизнедеятельности.                                                                         |
| <b>Р12</b>                          | Быть заинтересованным в непрерывном обучении и совершенствовании своих знаний и качеств в области электроэнергетики и электротехники.                                                                                                                                     |

1. Анализ существующих в ОИК СК-2007 средств обработки и подсистем дорасчета телеизмерений.
2. Анализ технологических задач, требующих дорасчета телеизмерений и телесигнализации
3. Формирование требований к разрабатываемому программному обеспечению и определение круга решаемых при помощи разрабатываемого ПО задач.
4. Разработка программы дорасчета и проверка ее на соответствие требованиям технического задания.

|                                                                                                    |                                                                                 |
|----------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------|
| <b>Перечень графического материала</b><br><i>(с точным указанием обязательных чертежей)</i>        | Блок-схемы алгоритма блинкерной фиксации, реализованного в программе дорасчета. |
| <b>Консультанты по разделам выпускной квалификационной работы</b><br><i>(с указанием разделов)</i> |                                                                                 |
| <b>Раздел</b>                                                                                      | <b>Консультант</b>                                                              |
| Финансовый менеджмент, ресурсоэффективность и ресурсосбережение                                    | Фигурко А. А.                                                                   |
| Социальная ответственность                                                                         | Невский Е.С.                                                                    |
| General information on processing of teleinformation in OIC                                        | Исакова Ю.И.                                                                    |
| <b>Названия разделов, которые должны быть написаны на русском и иностранном языках:</b>            |                                                                                 |
| Общие сведения об обработке телеинформации в ОИК                                                   |                                                                                 |
| General information on processing of teleinformation in OIC                                        |                                                                                 |

|                                                                                                 |          |
|-------------------------------------------------------------------------------------------------|----------|
| <b>Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику</b> | 08.02.17 |
|-------------------------------------------------------------------------------------------------|----------|

**Задание выдал руководитель:**

| Должность                         | ФИО                        | Ученая степень, звание | Подпись | Дата |
|-----------------------------------|----------------------------|------------------------|---------|------|
| Старший преподаватель кафедры ЭЭС | Васильев Алексей Сергеевич | К.Т.Н.                 |         |      |

**Задание принял к исполнению студент:**

| Группа | ФИО                     | Подпись | Дата |
|--------|-------------------------|---------|------|
| 5AM5Д  | Руденко Николай Юрьевич |         |      |

## **Реферат**

**Актуальность.** Актуальность проблемы определяется отсутствием в АО «СО ЕЭС» программного обеспечения для решения узкоспециализированных задач, связанных с проведением комплекса расчетов в реальном времени. Для проведения этих расчетов необходимо использование дополнительных алгоритмов, нереализуемых подсистемами дорасчета ОИК СК-2007.

**Цель и задачи исследования.** Целью настоящей работы является разработка программного модуля, позволяющего производить дорасчет параметров телеизмерений и телесигналов.

### **Задачи:**

1. Провести анализ существующих в ОИ СК-2007 средств обработки и подсистем дорасчета телеизмерений.
2. Провести анализ требований к разработке программного обеспечения, определить круг решаемых при помощи создаваемого программного обеспечения (ПО) задач.
3. Разработать программное обеспечение, реализующее основные алгоритмы дорасчета, чтение сохраненных алгоритмов алгоритм работы и интерфейс с базой данных реального времени ОИК СК-2007 программы.

**Методологическая основа.** При работе над диссертацией для проведения анализа использовалась нормативная документация о функционировании оперативно-информационного комплекса СК-2007. Разработка программы дорасчета выполнена на языке С#.

**Информационная база.** Поставленная цель достигается в процессе анализа принципов работы современных систем сбора, передачи, хранения и отображения информации, изучения специальной литературы, интернет источников, собственных исследований.

**Новизна исследования и научные результаты.** Предложен оригинальный способ создания и использования алгоритмов дорасчета параметров телеизмерений и телесигналов.

**Объектом исследования** магистерской диссертации являются программные средства дорасчета параметров телеизмерений и телесигналов.

**Предметом исследования** являются функциональные возможности программ дорасчета телеметрической информации.

**Апробация работы:** публикация материалов в сборниках VII международной молодежной научно-технической конференции «Электроэнергетика глазами молодежи – 2016» (г. Казань 19-23 ноября 2016).

**Практическая значимость.** Ожидаемым эффектом от внедрения разрабатываемого ПО является повышение эффективности разработки и реализации вычислительных алгоритмов, нереализуемыми по функциональным ограничениям в дорасчетах телеизмерений и телесигнализации для задач оперативно-информационного комплекса СК-2007 эксплуатируемого в АО «СО ЕЭС».

## **Основное содержание диссертации**

**Глава 1.** Глава посвящена рассмотрению организации обработки телеметрии в оперативно-информационном комплексе СК-2007. В главе рассмотрены существующие в комплексе виды дорасчетов, подсистема оперативного дорасчета obtits.exe, а также организация обмена информацией между клиентскими приложениями и базой данных реального времени ОИК.

**Глава 2.** Рассматриваются общие требования к разрабатываемому программному обеспечению и перечень задач, решаемых при помощи разработанного программного обеспечения (ПО). Формируются программные требования к разрабатываемому ПО.

**Глава 3.** Представлено описание структуры, элементов программы и алгоритма работы программы дорасчета параметров телеизмерений и телесигналов в среде программирования Microsoft Visual Studio на языке программирования C#. Приведен внешний вид интерфейса программы.

**Глава 4.** Посвящена вопросам финансового менеджмента, ресурсоэффективности и ресурсосбережения применительно к разработке программного обеспечения.

**Глава 5.** Производится анализ причин, приводящих к неправильному функционированию программы дорасчета параметров ТИ и ТС, вероятные последствия и возможные пути решения проблем.

## **Определения, обозначения, сокращения, нормативные ссылки**

В настоящей работе использованы ссылки на следующие стандарты:

ISO/IEC 25010:2011

ISO/IEC 23270

В работе использованы следующие обозначения и сокращения:

SQL - structured query language (язык структурированных запросов)

SWOT – Strengths, Weaknesses, Opportunities, Threats

XML – Extensible Markup Language (расширяемый язык разметки)

АО «СО ЕЭС» – акционерное общество «Системный оператор Единой энергетической системы»;

БДРВ – база данных реального времени;

КДД – компонент доступа к данным;

НИР – научно-исследовательская работа;

ОИК – оперативно-информационный комплекс;

ПО – программное обеспечение;

СО – Системный оператор;

ТИ – телеизмерения;

ТС – телесигналы



## Оглавление

|                                                                                       |    |
|---------------------------------------------------------------------------------------|----|
| Реферат .....                                                                         | 5  |
| Определения, обозначения, сокращения, нормативные ссылки .....                        | 8  |
| Введение.....                                                                         | 11 |
| Глава 1. Общие сведения об обработке телеинформации в ОИК СК-2007 ....                | 13 |
| 1.1. Организация дорасчета в ОИК СК-2007 .....                                        | 13 |
| 1.2. Обработка телемеханики средствами ОИК СК-2007 .....                              | 16 |
| 1.3. Взаимодействие ОИК с внешними программами.....                                   | 20 |
| 1.4. Выводы по главе 1 .....                                                          | 22 |
| Глава 2. Обоснование требований к разрабатываемому программному обеспечению .....     | 23 |
| 2.1. Особенности объектно-ориентированного программирования на языке С# .....         | 30 |
| 2.2. Требования к XML-документам.....                                                 | 38 |
| 2.3. Требования к программе дорасчета параметров ТИ и ТС.....                         | 44 |
| 2.4. Выводы по главе 2 .....                                                          | 45 |
| Глава 3. Разработка программы дорасчета параметров телеизмерений и телесигналов ..... | 46 |
| 3.1. Взаимодействие программного модуля с ОИК .....                                   | 46 |
| 1.2. Описание алгоритма работы программного модуля .....                              | 48 |
| 1.3. Фрагменты кода программы.....                                                    | 53 |
| 1.4. Выводы по главе .....                                                            | 54 |
| Глава 4. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение.....         | 56 |
| 4.1. SWOT-анализ .....                                                                | 57 |
| 4.2. Организация и планирование работ .....                                           | 60 |

|                                                                                                        |    |
|--------------------------------------------------------------------------------------------------------|----|
| 4.3. Расчет сметы затрат на исследование .....                                                         | 64 |
| 4.4. Оценка научно-технической результативности НИР .....                                              | 68 |
| 4.5. Выводы по главе 4 .....                                                                           | 70 |
| Глава 5. Социальная ответственность.....                                                               | 73 |
| 5.1. Причины, вызывающие некорректную работу программного обеспечения .....                            | 74 |
| 5.2. Причины, приводящие к неправильному функционированию ОИК СК-2007.....                             | 76 |
| 5.3. Причины, приводящие к неправильному функционированию программы дорасчета параметров ТИ и ТС ..... | 78 |
| 5.4. Выводы по главе 5 .....                                                                           | 79 |
| Заключение .....                                                                                       | 80 |
| Список используемых источников.....                                                                    | 81 |
| Приложение А .....                                                                                     | 83 |

## **Введение**

Нормальное функционирование энергосистемы в целом и отдельных энергообъектов принципиально невозможно без использования различных по назначению и исполнению систем управления, контроля и связи. Эти системы обеспечивают сбор, передачу, обработку и хранение всей необходимой для полноценного функционирования информации.

Информация играет ключевую роль в управлении энергосистемой. Помимо оперативной информации между объектами диспетчеризации, объектами диспетчеризации и диспетчерскими центрами, а также между диспетчерскими центрами происходит постоянный обмен оперативной и статистической информацией, необходимой для планирования деятельности предприятий и управления режимом энергосистемы.

В АО «СО ЕЭС» для решения подобных задач существуют собственные программные комплексы, ориентированные на выполнение различных задач. В частности, основным программным комплексом для приема, обработки, хранения и передачи телеметрической, отчетной и плановой информации о режиме работы энергетических объектов, сетей и систем является оперативно-информационный комплекс СК-2007.

Комплекс является универсальной системой для построения систем коммерческой и технологической диспетчеризации. ОИК СК-2007 состоит из комплекта независимых программ, выполняющих различные функции по оценке состояния, формированию отчетных документов, аналитических расчетов, загрузке данные из различных систем и прочее. Программой, выполняющей функции по обработке потока информации, поступающей из БДРВ по каналам телемеханики и межмашинного обмена, является программа обработки телемеханики obtits.exe.

Одной из функций этой программы является расчет формул оперативного дорасчета. В программе предусмотрен большой объем

доступных формул, предназначенных для решения практически всех поставленных перед программой задач. Однако иногда возникает необходимость в использовании формул и алгоритмов, не предусмотренных в программе obtits.exe. Поэтому для их реализации необходимо создание программы дорасчета.

## **Глава 2. Обоснование требований к разрабатываемому программному обеспечению**

Современные электроэнергетические сети являются сложными и разветвленными объектами, обладающими различными взаимовлияющими факторами. Поэтому решение вопросов связанных с проектированием, созданием, управлением и эксплуатацией объектов электроэнергетики требует использования математического вычислительного аппарата и вычислительной техники, а также систем связи и телекоммуникаций.

В электроэнергетике программное обеспечение предназначено для решения двух типов задач – проектных и эксплуатационных.

Проектные задачи – задачи по выбору параметров необходимого оборудования и проектируемой сети. Для решения подобных задач применяется специализированное программное обеспечение, осуществляющее расчет электроэнергетических параметров, а также системы автоматизированного проектирования электроустановок и подстанций.

Для решения задач эксплуатации Единой энергетической системы и качественного электроснабжения постоянно ведется оперативно-диспетчерское управление. Для надежного управления требуется использование большого количества различных систем сбора, обработки, хранения и передачи информации.

На данный момент на рынке предлагается значительное количество различных программных комплексов, обеспечивающих сбор, обработку, хранение и передачу информации. Изначально такие программные комплексы применялись только для сбора и отображения данных. В настоящее время такие программные комплексы могут оснащаться дополнительными модулями по расчету, анализу и представлению информации диспетчерскому персоналу.

Для моделирования различных по сложности аварийных ситуаций в энергосистеме и анализа действий обучаемого диспетчерского персонала при

их ликвидации, что является важной формой обучения, также обеспечивается специализированными тренажерными программными комплексами.

Обобщив задачи, решаемые при помощи программного обеспечения, можно представить их в схематичном виде. Схема приведена на рисунке 2



Рисунок 1 – Задачи, решаемые при помощи ПВК

Данное деление является весьма условным, так как разрабатываемое программное обеспечение постоянно совершенствуется. [3]

Существует несколько определений требования [4]:

- свойство программного обеспечения, обеспечивающие достижение пользователем поставленной задачи или решения проблемы;
- свойство программного обеспечения, которым должна обладать создаваемая программа или ее компонент, чтобы удовлетворить условиям, стандарта, спецификации, контракта или любой другой нормативной или формальной документации.

Другими словами, требование к программному обеспечению – то, что программа должна делать для пользователя, прибора или системы.

Для полного определения системы необходимо описать пять основных категорий элементов [4]:

- **Входные данные.** Необходимая информация о содержании входных данных, устройствах ввода, протоколах передачи.
- **Выходные данные.** Описание устройств вывода информации, содержание информации протоколы передачи.
- **Функции системы.**
- **Атрибуты системы.** Надежность, доступность, пропускная способность, удобство сопровождения.
- **Атрибуты системной среды.** Такие дополнительные требования, как способность функционировать в условиях операционных ограничений и нагрузок, а также совместимость с операционной системой.

Элементы приведены на рисунке 3.

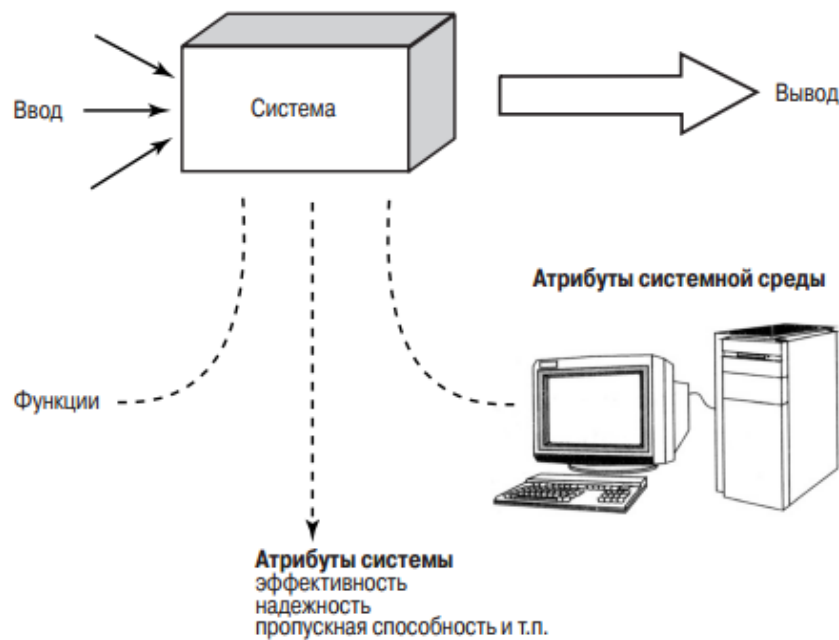


Рисунок 2 – Элементы системы

Если информация, представленная как требование к программе, не подходит ни под один из вышеуказанных типов информации, она не может считаться требованием.

К разрабатываемой программе могут предъявляться различные требования. Для предоставления обобщенного описания всех, выделяем их в нижеприведенные три типа:

- 1) функциональные требования к программному продукту;

- 2) нефункциональные требования к программному обеспечению;
- 3) ограничения проектирования.

Первые два типа объединены в группу программных требований.

Функциональные требования определяют, как будет вести себя система при определенных событиях, когда ей предоставлены необходимые входные данные или условия. Большая часть программных продуктов, предназначенных для выполнения различной работы, содержит большое количество функциональных требований. По большей части, основной целью разрабатываемого ПО является реализация именно функциональных требований.

При определении функциональных требований желательно придерживаться «золотой середины». Это подразумевает под собой соблюдение баланса между слишком обобщенным описанием и излишне конкретизированным описанием, при котором формулировка требования состоит из нескольких страниц текста. Наиболее оптимальным методом формулировки можно считать описание требований в виде простых декларативных определений, состоящих из нескольких предложений или в форме прецедентов. Подобное описание является наилучшим, так как при этом обеспечивается необходимый для разработчика уровень конкретизации [4].

Только функциональных требований недостаточно для полного описания системы. Поэтому необходимо учитывать требования называемые «нефункциональными». К ним относятся:

- практичность;
- надежность;
- производительность;
- возможность обслуживания.

Благодаря этой информации возможно больше узнать о создаваемой системе. Ниже рассмотрим каждый из этих пунктов подробнее.



## **Практичность**

Данное требование определяет, насколько легко будет изучаться программа потенциальными пользователями. Для удобства можно разбить пользователей на различные категории. К примеру: начинающие, имеющие опыт работы с программой, опытные и т.д.

Поскольку не существует точных критериев «практичности», при описании требования желательно их уточнять. Например, указывать время подготовки пользователя, не имеющего опыта работы с программой, для достижения минимально необходимой производительности. Для этого может потребоваться более конкретное описание «уровней» работы с программой.

Также можно задать в требовании время выполнения задач, осуществляемых пользователем. К примеру, программа для принятия и оплаты заказов в магазине. Для обслуживания покупателя пользователь (оператор/продавец) должен выполнить поиск запрашиваемого товара, внесение его в счет и оплату. В требовании возможно указать, сколько времени в среднем будет затрачиваться программой для обслуживания одного клиента.

Сравнение практичности проектируемой системы с уже существующей. Подобное задание требований является широко используемым методом. Однако необходимо учитывать возможность проверки соответствия. Требование необходимо описать так, чтобы пользователи могли проверить соответствие согласно прописанному критерию.

В данном требовании можно оговорить создание в программе системы подсказок, руководства пользователя, предупредительных оповещений и других различных форм вспомогательной документации.

## Надежность

В данном требовании необходимо указать, в какой степени система *должна* вести себя приемлемым для пользователя образом. Ниже приведены некоторые возможные аспекты:

- ***Доступность.*** Критерий указывает в течении какого времени программа должна быть доступна для использования. Доступность можно выразить в процентах на определенный период времени (99% доступность с 8:00 до 18:00). Полная доступность подразумевает под собой возможность использования всех функциональных возможностей программы в любое время.
- ***Среднее время между отказами.*** В большинстве случаев указывается в часах. Необходимо конкретное определение – что подразумевается под «отказами» программы.
- ***Среднее время восстановления.*** Название критерия говорит само за себя. Возможна конкретизация. Например: 80% процентов сбоев должны быть устранены в течение 5 минут. 99% - в течение часа после сбоя. Также необходима конкретизация определения «восстановление», подразумевает ли это полное восстановление всех функциональных возможностей, допускается ли частичное восстановление. Исходя из этого, можно задать время восстановления для каждого отдельного модуля программы.
- ***Точность.*** Необходимая числовая точность для выходных параметров. Например, точность до n-знаков после запятой, точность до копеек/рублей, пенни/долларов и т.д.
- ***Максимальный коэффициент ошибок.*** Число ошибок на тысячу строк кода или одну функцию.
- ***Количество различных ошибок.*** К примеру – разделение ошибок на малозначительные, некритичные и критические. Опять же необходима конкретизация для каждого определения.

## **Производительность**

К критериям производительности обычно относят следующее:

- Время ответа транзакции (среднее, максимальное)
- Пропускная способность (число транзакций в секунду)
- Емкость (какое количество пользователей/транзакций может обработать система)
- Допустимые режимы при снижении производительности

Если параллельно с программой должна происходить работа некоторого количества других программ, то может возникнуть необходимость указания в требовании используемых аппаратных ресурсов (память, ресурсы процессора и т.д.).

## **Возможность обслуживания**

Смысл данного требования заключается в указании способностей программного продукта к внесению изменений и модернизаций. В некоторых областях применения можно заранее предусмотреть природу изменений и указать необходимое время для внесения модификаций разработчиком.

## **Ограничения проектирования**

Так как ограничение проектирования не входит в вышеуказанные требования, то его можно не считать требованием. Однако если ограничение проектирования становится техническим, политическим или бизнес-условием и будет необходимо для соответствия требованиям контракта, стандарта, спецификации или другой документации, то оно становится требованием. В этом случае можно трактовать ограничение проектирования как любое другое требование. Однако лучше, чтобы подобных требований было как можно меньше, так как они могут серьезно ограничить возможности выбора или реализации решений.

Ограничения не влияют на поведение программы, но их необходимо выполнять ввиду контрактных, технических или деловых обязательств.

Источниками ограничений проектирования могут служить:

- язык программирования, операционная среда;
- совместимость с операционной системой;
- прикладные стандарты;
- корпоративные стандарты.

## **2.1. Особенности объектно-ориентированного программирования на языке C#**

Язык программирования C# – разработанный в 2001 году объектно-ориентированный язык программирования. Разработан как язык разработки приложений для платформы Microsoft .NET Framework. После этого был стандартизован как ECMA-334 и ISO/IEC 23270 [5].

По синтаксису этот язык наиболее близок к C++ и Java. Язык поддерживает полиморфизм, делегаты, перегрузку операторов, атрибуты, события и свойства, комментарии, статическую типизацию, исключения.

Несмотря на то что он перенял многое от своих языков-предшественников (C++, Pascal, Java Smalltalk), он исключает некоторые проблематичные при разработке программ модели. Например – C#, в отличие от C++ не поддерживает множественное наследование классов.

C# поддерживает все три «столпа» объектно-ориентированного программирования: инкапсуляцию, наследование и полиморфизм. Кроме того, в нем была реализована автоматическая «сборка мусора», обработки исключений, динамическое связывание.

Как уже было сказано выше, C# использует концепцию объектно-ориентированного программирования. Эта концепция является одной из ведущих на данный момент концепцией в области программирования.

Объектно-ориентированным программированием называется метод написания программ, основанный не на функциях и процедурах, а на объектах. Главным является объект а не действия, данные или логика. Результатами применения объектно-ориентированного программирования при написании программы являются процессы упрощение организации и создания структуры программы. А тот факт, что возможна замена отдельных объектов без воздействия на остальные, упрощает процедуру внесения изменений.

Двумя ключевыми понятиями объектно-ориентированного программирования являются понятия *класса* и *объекта* [6].

**Класс** – абстрактный тип данных, описывающий объект. Содержит данные и методы обработки этих данных;

**Объект** – конкретный экземпляр класса. Существует в памяти и содержит как данные, так и методы их обработки.

Объектно-ориентированное программирование основывается на некоторых базовых принципах [7]:

- **Инкапсуляция.** Применение этого принципа позволяет скрыть внутреннюю реализацию класса. Внутри класса возможно реализовать различные вспомогательные методы и определить вопросы доступа к ним и их видимости. Идея инкапсуляции тесно связана понятием защиты данных.

- **Наследование.** Самый популярный принцип. Позволяет создавать класс на основе другого. Под этим подразумевается возможность повторного использования кода – если уже существует класс, в котором уже прописана необходимая логика и функции, то есть возможность включить имеющийся класс (базовый) в новый (наследующийся).

- **Полиморфизм.** «Поли» - «много», «морфизм» - «изменение». Как следует из названия принципа, он позволяет одним и тем же объектам принимать различные формы, иметь различную реализацию. Язык программирования поддерживает полиморфизм, если классы с одинаковой спецификацией могут иметь различную реализацию — например, реализация

класса может быть изменена в процессе наследования. Кратко смысл полиморфизма можно выразить фразой: «Один интерфейс, множество реализаций». Полиморфизм позволяет писать более абстрактные программы и повысить коэффициент повторного использования кода.

- **Абстракция.** Принцип позволяет скрывать внутреннюю логику сущностей от конечного пользователя и выделять только необходимые для описания объекта характеристики и методы.

При разработке программы с использованием методов объектно-ориентированного программирования на языке C# средствами Microsoft Visual Studio необходимо решить следующие задачи:

- Определить необходимые для решения задачи классы;
- Построение алгоритма реализации основного действия в задаче;
- Создать необходимый для решения задачи математический аппарат;
- Осуществить программную реализацию с использованием языка C#.

Существует международный стандарт ISO/IEC 25010:2011 который различает восемь различных характеристик программного обеспечения [8]:

- обслуживаемость;
- функциональная пригодность;
- совместимость;
- эффективность работы;
- удобство использования;
- надежность;
- безопасность;
- переносимость.

При создании программы регулярно требуется вносить изменения в программный код, что может быть сложным при большом объеме последнего

или в случае, когда код писал другой программист. Критерий сложности внесения изменений можно назвать *обслуживаемость*.

Для программного обеспечения не характерны такие определения физического оборудования как поломка или износ, так как оно не является физической сущностью и не склонна к ним. Однако для поддержания полноценной работы программы после ввода в эксплуатацию необходима постоянная поддержка и модернизация. Это называется обслуживанием программного обеспечения. Существует четыре вида обслуживания ПО:

- корректирующее обслуживание. Устранение найденных ошибок;
- адаптивное обслуживание. Адаптация программы к изменениям окружающей среды. Примером изменения среды может служить смена операционной системы;
- улучшающее обслуживание. Выполнение новых требований пользователей программного обеспечения и/или заказчиков;
- профилактическое обследование. Повышение качества ПО и предотвращение сбоев и ошибок программы.

Для облегчения обслуживания программного кода, написанного на языке C# можно привести несколько рекомендаций [9].

### **Написание исходного кода короткими блоками**

Под блоками на языке C# можно понимать методы и конструкторы. Это небольшие фрагменты кода, которые можно выполнять и редактировать изолированно от других. Короткий блок может хранить и выполнять только одну функцию, длинный – несколько. Можно выделить три преимущества подобной реализации кода: простота тестирования, анализа и повторного использования.

Компилятор C# не обнаруживает ошибки связанные с логикой, их можно найти только при помощи анализа кода. Написание программы

короткими блоками позволяет, применяя различные методы тестирования, быстро найти некорректно функционирующий блок. И в найденном блоке, содержащем обычно около 15 строк кода, логические ошибки можно легко найти при помощи анализа кода.

Для упрощения программы каждый блок может быть использован неоднократно вызовом из разных методов. Длинные же блоки обычно заключают в себе специфические детали и специфические возможности, поэтому их повторное применение затруднено.

### **Написание кода простыми блоками**

Под простым блоком можно понимать блок, содержащий не более 4 точек ветвления. Точка ветвления, это оператор, который задает несколько направлений выполнения программы. В С# примерами точек ветвления могут быть операторы *if* и *switch*. Чем больше таких операторов в блоке кода, тем он сложнее.

Точки ветвления можно подсчитать во всей базе кода, классах, пакетах или блоках кода. Число точек ветвления в блоке кода равно минимальному числу путей, необходимых для охвата всех ветвей, созданных этими точками ветвления. Это называется охватом ветвей. Однако при учете всех путей, от первой строки до конечного оператора, может возникать комбинаторный эффект. Причина в том, что может иметь значение порядок прохождения ветвей. Все возможные комбинации ветвей называются путями выполнения, то есть они определяют максимальное количество путей через блок.

Точками ветвления в С# являются такие операторы и инструкции как:

- `if;`
- `case;`
- `?, ??;`
- `&&, |;`
- `while;`



- for, foreach;
- catch.

На рисунке 4 рассматривается блок, включающий в себя два последовательных оператора *if*. Рисунок отображает разницу между охватом ветвей и путей выполнения. Охват ветвей выполнен сплошными линиями, пути выполнения – пунктирными.

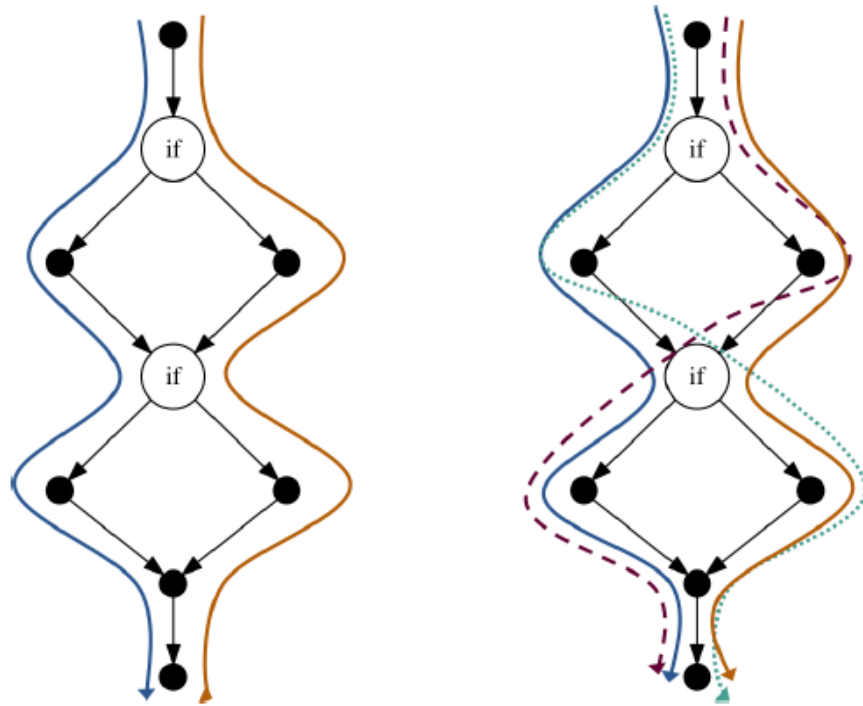


Рисунок 3 – Охват ветвей и путей выполнения

Написание кода простыми блоками дает два важных преимущества: простота тестирования, поскольку количество тестов соответствует количеству ветвей, и простота изменения. Блоки кода с высокой сложностью трудно понять, что приводит к проблемам при их изменении.

### Исключать дублирование кода

Дублирование кода является одним из основных источников регрессивных ошибок, которые приводят к тому, что код, нормально функционирующий до определенного момента, перестает работать. Дублированными фрагментами кода являются идентичные участки кода длиной не менее шести строк без учета комментариев и пустых строк.

Если два участка кода совершенно одинаковы (как текстуально, так и синтаксически), они называются *клонами первого вида*. Если два фрагмента кода одинаковы лишь синтаксически – они являются *клонами второго вида*. Эти клоны отличаются друг от друга лишь комментариями, пробелами, названиями идентификаторов и значениями литералов.

Эта рекомендация относится только к клонам первого вида, так как:

- улучшение обслуживаемости происходит в основном при удалении клонов именно первого вида;
- клоны первого вида намного проще обнаружить как программными средствами, так и «ручным» анализом.

В любом коде так или иначе могут появляться ошибки. Однако если ошибка будет содержаться в дублированном участке, то она повторится несколько раз. Из этого вытекает и необходимость многократного исправления одной и той же ошибки, что приводит к лишним тратам времени и увеличивает вероятность допущения новой ошибки. Поэтому, если возникает необходимость в повторном использовании фрагмента, его рекомендуется поместить в отдельный метод и при необходимости вызывать из другого метода.

### **Разделение задач на модули**

Приведенные выше рекомендации относятся к рекомендациям для блоков кода и касаются упрощения отдельных методов и конструкторов. Рекомендация по разделению задач относится уже к уровню модулей. В объектно-ориентированном языке программирования понятию *модуля* соответствует *класс*. Эта рекомендация направлена на установление гибких связей между классами и упрощение анализа и внесения изменений в программу.

Преимуществом маленьких классов является слабая связь между ними. Она обеспечивает большую гибкость на уровне классов, что облегчает внесение изменений. Под «гибкостью» понимается именно возможность

внесения изменений при небольшом риске появления неожиданных ошибок. Таким образом, слабая связь позволяет работать с отдельными частями кода, не оказывая значительного влияния на остальной код. Когда класс сильно связан с другими, то внесение в него изменений повлияет на всю систему в целом. В этом случае, помимо необходимости внесения изменений во все связанные классы, увеличивается риск появления ошибок и несоответствий в отдаленных местах системы.

Напротив, обеспечение гибкости кода облегчает анализ, поскольку классы инкапсулируют данные и реализацию для решения одной задачи. Подобно тому, как можно было давать понятные имена методам, ответственным за что-то одно, классы, если они используются для достижения какой-то одной цели, также можно называть так, чтобы было сразу ясно, для чего они предназначены. Уверенность в том, что один класс отвечает только за что-то одно, носит название принципа персональной ответственности.

### **Избегать тесных связей между элементами архитектуры**

Под слабой зависимостью понимается возможность изолированного внесения изменений. Изолированное обслуживание уменьшает объем работы, поскольку изменения в коде не затрагивают ничего находящегося за его пределами. Слабая связь обеспечивается за счет ограничения количества точек доступа и объема информации, используемой компонентами совместно.

Связанность модулей определяется объемом функций, экспортируемых отдельными модулями (классами) остальному коду. Связанность компонентов определяется количеством модулей, экспортируемых одним компонентом (группой модулей) другому. Если модуль вызывает другой модуль в том же компоненте, то вызов считается внутренним на уровне компонентов, но считается связанным на уровне модулей.

Если все компоненты независимы друг от друга, то это упрощает распределение ответственности за их обслуживание. Это является необходимым условием при разделении разработчиков на команды или между членами одной команды. Однако, если компоненты тесно связаны, ситуация становится полностью противоположной. Это влечет за собой усложнение процесса тестирования программы, увеличение времени для согласования действий разработчиков и в целом ухудшит координацию.

## **2.2. Требования к XML-документам**

На данный момент существует большое обилие различных баз данных, из-за чего возникает проблема при переносе данных из одного типа в другой. Для решения этой проблемы могут использоваться программы-конвертеры, осуществляющие преобразование одного формата баз данных в другой. Однако, такие программы существуют только для самых распространенных форматов баз данных, что создает проблему при переносе данных устаревшего формата. Использовать «каскад» конвертеров или создавать новый нерационально, поэтому одно из самых оптимальных решений - хранить базы данных в XML-файлах в качестве промежуточного этапа.

XML – (англ. eXtensible Markup Language, расширяемый язык разметки) представляет собой язык разметки, который был создан для описания данных. Представлен набором символов или последовательностей, вставляемых в текст для передачи информации о его выводе или строении. Документ, созданный при помощи такого языка, содержит не только текст, но и дополнительную информацию о его структуре. Кроме того, язык разметки позволяет вставлять в документ интерактивные элементы и содержание других документов [10].

Язык XML был создан для обмена, транспортировки и хранения данных. С его помощью можно реализовать обмен данных между различными системами.

XML-файлы и файлы других расширений, основанных на языке XML получили широкое распространение, практически каждая современная операционная система поддерживает этот формат файлов. В XML-файлах могут храниться самые различные данные – от фрагментов баз данных до настроек программного обеспечения. Данный язык разметки изначально был задуман для обмена информацией между программами в сети Интернет. Так как файлы XML представлены в текстовом виде, их можно редактировать или менять кодировку данных при помощи любого текстового редактора.

#### Преимущества языка XML:

1. Формат языка является самоописывающим (разметка языка описывает данные). Это снимает ограничение на количество тегов, которыми можно описать хранимую информацию;
2. Другим достоинством является возможность использования XML в качестве универсального языка запросов к хранилищам информации;
3. Язык XML может выступать в качестве уникального способа хранения данных, который содержит в себе как средства для разбора информации, так и средства для представления ее клиенту.
4. XML позволяет проводить проверку корректности данных, хранящихся в документах, производить проверки иерархических соотношений внутри документа и устанавливать единый стандарт на структуру документов. Это дает возможность его использования при построении сложных информационных систем, в которых очень важным является вопрос обмена информацией между различными приложениями, работающими в одной системе.

#### Недостатки XML:

1. Отсутствие многих возможностей, присущих современным базам данных: экономичность хранения, безопасность, транзакции, многопользовательский доступ, запросы по многим документам и др.
2. Многословность языка и как следствие – медленный доступ к данным из-за необходимости парсирования и конвертирования текста.

На основании вышесказанного можно сделать вывод, что использование XML документов непрактично в базах данных, описывающих бизнес-среды, в которых действуют строгие требования к интегрированности данных, высокие требования к производительности и существует необходимость работы с большим количеством пользователей. Однако из этого же следует что использование XML-документов в качестве базы данных в средах с небольшим количеством данных, пользователей и невысокими требованиями к производительности обосновано. Наиболее простым примером может служить база данных телефонных номеров или каталог библиотечных книг.

### ***Синтаксис языка XML***

Пример XML-документа:

```
<?xml version = "1.0" encoding = "utf-8"?>
<book>
  <title>Принципы работы с требованиями к программному обеспечению
</title>
  <author> Дин Леффингуэлл </author>
  <year>2002</year>
  <pages>448</pages>
</book>
```

Документ XML может располагаться в одном или нескольких файлах, причем часть из них может находиться на разных компьютерах. Для интеграции содержимого разных файлов в один объект в XML используется специальная разметка, который можно охарактеризовать как логическую структуру.

Как правило, XML-документ содержит следующие разделы:

- XML-декларация
- Пролог;

- Элементы;
- Атрибуты;
- Комментарии.

### ***Декларация***

XML-декларация расположена в первой строке XML-документа. XML-декларация не является обязательной и в схеме документа состоит из следующих элементов:

- *Номер версии:*

<?xml version="1.0"?>.

Обязательный аргумент. Текущая версия —1.0.

- *Декларация кодировки:*

<?xml version="1.0" encoding="UTF-8"?>

Необязательный параметр. Декларация кодировки содержит значение, представляющее собой существующую кодировку символов.

- *Декларация автономности:*

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>.

Как и декларация кодировки – необязательный параметр. При использовании должна располагаться на последнем месте в XML-декларации.

### ***Пролог***

Прологом называются данные, расположенные после открывающего тега документа или после корневого элемента. Он включает сведения, относящиеся к документу такие как: кодировка символов, структура документа, таблицы стилей.

### ***Элементы***

Элементы в XML документе отвечают за организацию информации и являются основными структурными единицами языка XML. Они образуют

структуры, которые легко обрабатываются программно или при помощи таблиц стилей. Элементы оформляются следующим образом:

**<title>**Принципы работы с требованиями к программному обеспечению  
**</title>**

**<author>** Дин Леффингуэлл **</author>**

**<year>**2002**</year>**

**<pages>**448**</pages>**

Каждый элемент состоит из начального тега **<title>**, конечного тега **</title>** и его содержимого – XSLT. Теги устанавливают границы содержимого элемента.

Особенностью языка XML является то, что элементы могут содержать в себе другие элементы или конструкции языка. Тем самым появляется возможность формирования иерархической структуры в виде дерева. На рисунке 5 приведен пример подобной структуры

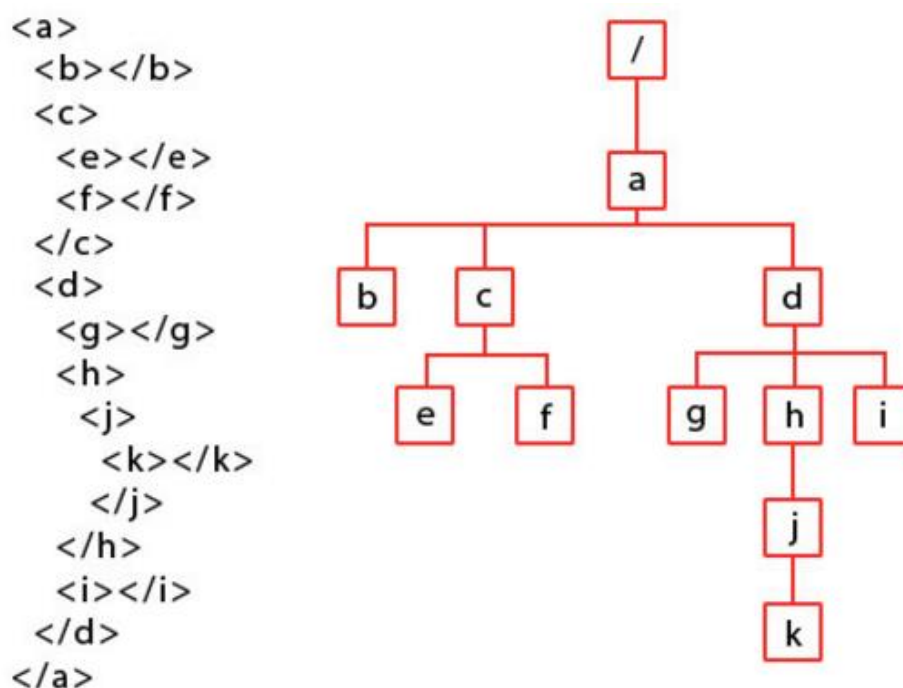


Рисунок 4 – Древовидная структура XML

Эта организация документа является простой структурой для обработки, однако при этом позволяет описывать достаточно сложные конструкции.



## ***Атрибуты***

В XML элементы могут содержать атрибуты. Атрибуты позволяют добавлять сведения об элементе с помощью пар «имя-значение». Атрибуты часто используются для определения тех свойств элементов, которые не считаются содержимым элемента. Атрибуты могут отображаться в открывающих или пустых тегах, но не в закрывающих тегах. Атрибут для элемента задается следующим образом:

```
<Element1 attribut1="value" ></Element1>
```

Атрибуты должны иметь имя и значение. В одном элементе не могут быть два атрибута с одинаковыми именами.

## ***Комментарии***

Содержимое, не предназначенное для синтаксического анализа (например, замечания о структуре документа или редактировании), можно заключить в комментарии. Комментарии применяются в том случае, если необходимо оставить какие-либо пояснения в документе, либо временно исключить часть документа из обработки.

Комментарии начинаются с группы символов <!-- и заканчиваются группой символов -->:

```
<!--Текст комментария -->
```

Однако существуют два правила написания комментариев:

- В тексте комментария не разрешается написания двух символов «-» подряд;
- Комментарий не может заканчиваться символом «-».

### 2.3. Требования к программе дорасчета параметров ТИ и ТС

На основании вышеприведенных информации можно составить список требований к разрабатываемой программе.

#### **Функциональные требования:**

1. Входными данными для программы должны являться телеизмерения и телесигналы, поступающие от базы данных реального времени оперативно-информационного комплекса.
2. Выходными данными также являются ТИ и ТС, записываемые в БДРВ ОИК;
3. Программа должна производить дорасчет в реальном времени, то есть все функции должны быть рассчитаны за 1 секунды (поскольку количество дорасчетов заранее неизвестно, то лучше задать количество выполняемых расчетов в секунду);
4. Наименования входных и выходных ТИ, ТС, а также дорасчетные функции и порядок их выполнения должны быть получены из предоставленного XML-файла.

#### **Нефункциональные требования:**

**Практичность.** Для работы с программой должно быть достаточным иметь начальные навыки работы с персональным компьютером.

**Надежность.** Программа должна быть доступна для работы в любое время. И иметь минимально возможное время для восстановления работы.

#### **Ограничения программирования.**

1. Программа должна быть реализована на платформе Microsoft Visual Studio на языке программирования C#;
2. Алгоритм дорасчета должен храниться в конфигурационном файле в формате XML.

## **2.4. Выводы по главе 2**

1. Не все запросы, предъявляемые к разрабатываемому программному обеспечению, могут считаться требованиями. Для того, чтобы считаться требованием, определение должно удовлетворять одному из двух условий: быть свойством ПО, обеспечивающим достижение поставленной задачи или выражать свойство программы, которое необходимо для удовлетворения условий контрактов или нормативной документации.

2. Разработка программного модуля осуществляется на языке C#, так как является простым, универсальным и удобным в использовании объектно-ориентированным языком программирования.

3. Так как язык XML является расширяемым языком разметки и может хранить не только информацию, но и сведения о структуре документа. Это позволяет использовать его для обмена данными между различными системами, что в свою очередь позволяет применять XML документы для хранения и передачи дорасчетных алгоритмов.

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА  
«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И  
РЕСУРСОСБЕРЕЖЕНИЕ»**

Студенту:

|               |                         |
|---------------|-------------------------|
| <b>Группа</b> | <b>ФИО</b>              |
| 5AM5Д         | Руденко Николай Юрьевич |

|                     |              |                           |                                                                                   |
|---------------------|--------------|---------------------------|-----------------------------------------------------------------------------------|
| <b>Институт</b>     | <b>ЭНИН</b>  | <b>Кафедра</b>            | <b>ЭЭС</b>                                                                        |
| Уровень образования | магистратура | Направление/специальность | Автоматизированные системы диспетчерского управления электроэнергетических систем |

**Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:**

|                                                                                                                                                |                                                      |
|------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------|
| 1. Стоимость ресурсов при разработке программного продукта: материально-технических, энергетических, финансовых, информационных и человеческих | Стоимость материалов и должностные оклады            |
| 2. Нормы и нормативы расходования ресурсов                                                                                                     | Норма амортизации                                    |
| 3. Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования                                            | Отчисления во внебюджетные фонды<br>Налог на прибыль |

**Перечень вопросов, подлежащих исследованию, проектированию и разработке:**

|                                                                                                                                                       |                                                           |
|-------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------|
| 1. Оценка коммерческого потенциала, перспективности и альтернатив разработки программного продукта с позиции ресурсоэффективности и ресурсосбережения | Оценки НТУ                                                |
| 2. Планирование и формирование бюджета на разработку программного продукта                                                                            | Расчёт текущих затрат на разработку программного продукта |
| 3. Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности разработки                             | Оценка научно-технического уровня НИР                     |

**Дата выдачи задания для раздела по линейному графику**

**Задание выдал консультант:**

|                  |                             |                               |                |             |
|------------------|-----------------------------|-------------------------------|----------------|-------------|
| <b>Должность</b> | <b>ФИО</b>                  | <b>Ученая степень, звание</b> | <b>Подпись</b> | <b>Дата</b> |
| Доцент           | Фигурко Аркадий Альбертович | к.э.н.                        |                |             |

**Задание принял к исполнению студент:**

|               |                         |                |             |
|---------------|-------------------------|----------------|-------------|
| <b>Группа</b> | <b>ФИО</b>              | <b>Подпись</b> | <b>Дата</b> |
| 5AM5Д         | Руденко Николай Юрьевич |                |             |

#### **Глава 4. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение**

Для определения и анализа трудовых и денежных затрат, направленных на реализацию проекта, а также с целью определения уровня их научно-технической результативности проводится технико-экономическое обоснование научно-исследовательских работ.

Целью раздела «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение» является определение перспектив и успешности научно-исследовательской работы, разработка механизмов управления и сопровождения проектных решений на этапах реализации.

В данном проекте производилась разработка программного комплекса осуществляющего дорасчет ТИ и ТС, поступающих с объектов диспетчеризации СО ЕЭС. Разработка производилась в программной среде Microsoft Visual Studio C#. Второстепенные операции проводились при помощи программного комплекса Microsoft SQL Server.

Ожидаемый результат работы:

1. Разработка программного комплекса для дорасчета ТИ, ТС в программной среде C#;
2. Повышение навыков работы с ЭВМ, рост квалификации инженерно – технических работников
3. Решение задачи выполнения дорасчетных алгоритмов, полученных из конфигурационного XML-файла.

В данной части проекта осуществляется определение этапов работы, составление план-графика, сметы затрат. План–график составляется для того, чтобы выполнить научно- исследовательскую работу в срок при наименьших затратах средств. В нем содержится информация о поэтапной трудоемкости работ.

Для определения трудоемкости выполнения проекта составим перечень основных этапов и видов работ, которые должны быть выполнены.

#### **4.1. SWOT-анализ**

SWOT – Strengths (сильные стороны), Weaknesses (слабые стороны), Opportunities (возможности) и Threats (угрозы) – является комплексным способом анализа проекта. SWOT-анализ используется при проведении исследований внешней и внутренней среды проекта [12].

На первом шаге анализа описываются сильные и слабые стороны проекта, выявляются возможности и угрозы при его реализации.

К сильным сторонам относятся факторы, определяющие конкурентоспособность проекта. Существование у проекта сильных сторон указывает на его отличительные преимущества или на наличие особых ресурсов, которые имеются в распоряжении руководства проекта и которые можно эффективно использовать для достижения поставленной цели. Сильные стороны необходимо рассматривать как с точки зрения руководителей проекта, так и с точки зрения тех, кто задействован в его выполнении.

Слабые стороны – недостатки или упущения проекта, наличие которых может препятствовать достижению поставленных целей. Они могут выражаться в недостатке возможностей или ресурсов в сравнении с конкурентами.

Возможности заключаются в любых предпочтительных ситуациях в настоящем или будущем, возникающих в окружающей среде проекта и состоят в тенденциях, изменениях или предполагаемых потребностях, которые служат для поддержания спроса на результат проекта и позволяют его руководству повысить конкурентоспособность. Примерами таких ситуаций могут быть изменения качества жизни населения, политики государства по отношению к отрасли и т.д.

К угрозам относятся любые нежелательные ситуации, тенденции или изменения, которые могут оказывать разрушительный или угрожающий характер конкурентоспособности проекта. Угрозами могут выступать барьеры или ограничения, способные нанести вред проекту, например, изменение в законодательстве, снижение или задержки финансирования и т.д.

Список возможностей, угроз, а также сильных и слабых сторон проекта приведен в таблице 1.

Таблица 1 –SWOT-анализ научно-исследовательского проекта

| Сильные стороны проекта                                                                                                                                                                                                                            | Слабые стороны проекта                                                                                                                                                                                 |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| С1. Применение новых технологий.<br>С2. Простота эксплуатации.<br>С3. Возможность внесения модификаций для расширения функционала.<br>С4. Компетентность руководства.<br>С5. Низкие требования к оборудованию.<br>С6. Низкая стоимость разработки. | Сл1. Узкая специализация.<br>Сл2. Несовместимость с другими программными продуктами.<br>Сл3. Уязвимость по отношению к конкурентному давлению.<br>Сл4. Отсутствие собственного графического редактора. |
| Возможности                                                                                                                                                                                                                                        | Угрозы                                                                                                                                                                                                 |
| В1. Появление дополнительного спроса на продукт.<br>В2. Возможность обслуживания новых групп потребителей.<br>В3. Отсутствие зарубежных конкурентов.<br>В4. Ослабление нестабильности бизнеса.                                                     | У1. Возможность появления новых конкурентов.<br>У2. Возрастающее конкурентное давление.<br>У3. Рост требований потребителей.<br>У4. Появление государственных требований к сертификации продукции.     |

Далее интерактивная матрица проекта. С ее помощью осуществляется выявление различных комбинаций взаимосвязей областей матрицы SWOT. В ней каждый фактор помечается знаками «+» - указывает на сильное соответствие сильным возможностям, «-» - указывает на слабое соответствие, в случае неопределенности ставится «0».

Интерактивная матрица проекта представлена в таблице 2.

Таблица 2 – Интерактивная матрица проекта

| Сильные стороны проекта |    |    |    |    |    |    |    |
|-------------------------|----|----|----|----|----|----|----|
| Возможности проекта     |    | C1 | C2 | C3 | C4 | C5 | C6 |
|                         | B1 | +  | +  | 0  | -  | +  | -  |
|                         | B2 | +  | +  | +  | -  | +  | -  |
|                         | B3 | 0  | +  | +  | 0  | -  | -  |
|                         | B4 | -  | -  | -  | +  | +  | 0  |

Анализ интерактивной матрицы позволяет выделить два основных пути реализации проекта: B2C1C2C3C5 и B1C1C2C5.

Возможные способы коррекции слабых сторон проекта и борьбы с угрозами приведены ниже.

Сл1. Расширение функционала программного обеспечения.

Сл2. Расширение списка поддерживаемых форматов описания дорасчетных алгоритмов.

Сл3. Привлечение различных компаний, заинтересованных в создании данного программного продукта.

Сл4. Разработка способа редактирования графического алгоритма дорасчета.

У1. Проведение рекламных компаний.

У2. Поиск потенциальных спонсоров и их привлечение.

У3. Повышение качества программного продукта, расширение функционала.

У4. Поиск путей выхода на зарубежные рынки.



Очевидным недостатком разрабатываемого программного обеспечения является его узконаправленность, поскольку число технологических процессов, которые целесообразно реализовывать в данном программном продукте - мало.

Сложность выхода на рынок, в связи с узконаправленной средой применения ПО, также является неотъемлемой особенностью вывода на рынок продуктов данного типа.

#### **4.2. Организация и планирование работ**

При организации процесса реализации конкретного проекта необходимо оптимально планировать занятость каждого из его участников и сроки проведения отдельных работ.

На первом этапе планирования составляется список проводимых работ и определяются их исполнители. Результат планирования может быть представлен диаграммой Ганта. Диаграмма Ганта – столбчатая диаграмма (гистограмма), которая используется для иллюстрации плана, графика работ по какому-либо проекту. По сути, диаграмма Ганта состоит из полос, ориентированных вдоль оси времени. Каждая полоса на диаграмме представляет отдельную задачу в составе проекта (вид работы), её концы — моменты начала и завершения работы, её протяженность — длительность работы. Вертикальной осью диаграммы служит перечень задач. Диаграмма Ганта не отображает значимости или ресурсоемкости работ. Для крупных проектов диаграмма Ганта становится чрезмерно тяжеловесной и теряет всякую наглядность. Указанные недостатки серьёзно ограничивают область применения диаграммы. Тем не менее, в настоящее время диаграмма Ганта является стандартом де-факто в теории и практике управления проектами, по крайней мере, для отображения Структуры перечня работ по проекту.

Перечень задач и подзадач, а также их исполнители (научный руководитель – НР, инженер - И) приведены в таблице 3.

Таблица 3 – Перечень работ

| <b>Код работ</b> | <b>Название</b>                                                           | <b>Длительность, дни</b> | <b>Дата начала работ</b> | <b>Дата окончания работ</b> | <b>Исполнители</b> |
|------------------|---------------------------------------------------------------------------|--------------------------|--------------------------|-----------------------------|--------------------|
| A                | Выбор актуальной темы                                                     | 4                        | 1.09.2015                | 5.09.2015                   | НР                 |
| B                | Составление задания                                                       | 6                        | 6.09.2015                | 12.09.2015                  | НР                 |
| C                | Получение задания                                                         | 1                        | 13.09.2015               | 14.09.2015                  | И                  |
| D                | Поиск и изучение литературы                                               | 20                       | 15.09.2015               | 5.10.2015                   | И                  |
| E                | Выбор направления исследования                                            | 5                        | 6.10.2015                | 11.10.2015                  | НР,И               |
| F                | Описание системы сбора информации об аварийных нарушениях                 | 91                       | 11.10.2015               | 11.01.2016                  | И                  |
| G                | Написание отчета                                                          | 30                       | 12.01.2016               | 12.02.2016                  | И                  |
| H                | Исследование нормативной документации                                     | 92                       | 13.02.2016               | 17.05.2016                  | И                  |
| I                | Исследование схожих по своей спецификации программных модулей и продуктов | 141                      | 18.05.2016               | 1.10.2016                   | И                  |
| J                | Разработка программного модуля для дорасчета телеизмерений и телесигналов | 181                      | 2.10.2016                | 1.04.2017                   | И                  |
| K                | Сопровождение и отладка разработанного программного модуля.               | 30                       | 2.04.2017                | 1.05.2017                   | И                  |
| L                | Составление раздела социальной ответственности и финансового менеджмента  | 18                       | 2.05.2017                | 20.05.2017                  | И,К                |
| M                | Оформление пояснительной записки                                          | 11                       | 21.05.2017               | 1.06.2017                   | И                  |
| N                | Подготовка к защите                                                       | 13                       | 2.06.2017                | 15.06.2017                  | И                  |
| <b>Итого</b>     |                                                                           | <b>643</b>               |                          |                             |                    |

Перечень планируемых работ в виде диаграммы Ганта представлен на рисунке 9.

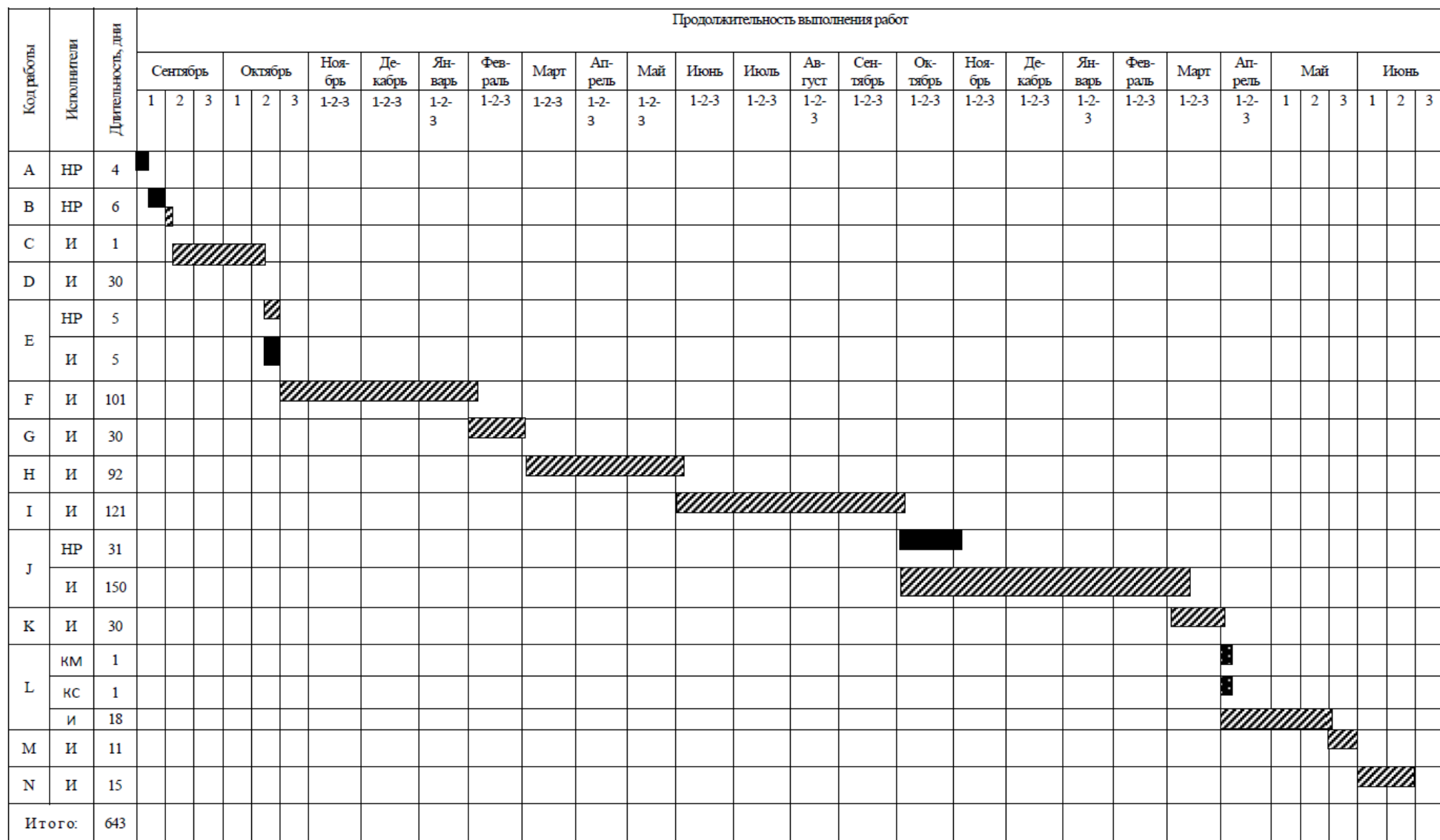


Рисунок 5 – Диаграмма Ганта

#### 4.3. Расчет сметы затрат на исследование

В состав затрат на создание проекта включается стоимость всех расходов, необходимых для реализации комплекса работ, составляющих содержание данной разработки[12]. Расчет сметной стоимости на выполнение данной разработки производится по следующим статьям затрат:

- материалы и покупные изделия;
- основная заработная плата;
- дополнительная заработная плата
- отчисления в социальные фонды;
- расходы на электроэнергию;
- амортизационные отчисления;
- работы, выполняемые сторонними организациями;
- прочие расходы.

К расчету затрат на материалы относится стоимость материалов, покупных изделий, полуфабрикатов и других материальных ценностей, расходуемых непосредственно в процессе выполнения работ. Цена материальных ресурсов определяется по средней рыночной стоимости на 2015 год по соответствующим ценникам и приведена в таблице 4.

Таблица 4 – Расходные материалы

| Наименование материалов | Цена за ед., руб. | Количество | Сумма, руб. |
|-------------------------|-------------------|------------|-------------|
| Офисная техника         |                   |            |             |
| Компьютер               | 22000             | 1          | 22000       |
| Клавиатура              | 700               | 1          | 700         |
| Мышка                   | 600               | 1          | 600         |
| Принтер                 | 5000              | 1          | 5000        |
| Офисная мебель          |                   |            |             |
| Стол                    | 3000              | 1          | 2000        |
| Стул                    | 700               | 1          | 700         |
| <b>Итого</b>            | <b>31000</b>      |            |             |

Расходы на материалы составили  $S_{\text{мат}} = 31000$  руб.

Расчет основной заработной платы включает заработную плату научного руководителя и инженера, а также премии, входящие в фонд

заработной платы. Расчет основной заработной платы выполняется на основе трудоемкости выполнения каждого этапа и величины месячного оклада исполнителя. Среднедневная заработная плата рассчитывается по формуле:

$$\text{Дневная з/п} = \frac{\text{Месячный оклад}}{20,58 \text{ дней}}$$

Расчеты затрат на основную заработную плату приведены в таблице 5.

При расчете учитывалось, что в году 247 рабочих дня и, следовательно, в месяце 20,58 рабочих дня. Затраты времени на выполнение работы по каждому исполнителю брались из таблицы 7. Также был принят во внимание коэффициент, учитывающий коэффициент по премиям  $K_{\text{пр}} = 0,1$  и районный коэффициент  $K_{\text{рк}} = 0,3$ .

$$K = 1 + K_{\text{пр}} + K_{\text{рк}} = 1 + 0,1 + 0,3 = 1,4$$

Таблица 5 – Затраты на основную заработную плату

| Исполнитель   | Оклад, руб./мес | Месячная з/п. работника, руб. | Средне-дневная ставка, руб./день | Затраты времени, дни | Коэффициент | Фонд з/платы, руб. |
|---------------|-----------------|-------------------------------|----------------------------------|----------------------|-------------|--------------------|
| НР            | 10223,33        | 14312,66                      | 695,46                           | 35                   | 1,4         | 24341,26           |
| И             | 14874,65        | 20824,51                      | 1011,88                          | 630                  | 1,4         | 637484,4           |
| <b>Итого:</b> |                 |                               |                                  |                      |             | <b>661826</b>      |

Таким образом, затраты на основную заработную плату составили

$$C_{\text{осн}} = 661826 \text{ (руб.)}$$

Затраты по статье отчислений от заработной платы составляют отчисления во внебюджетные фонды.

Отчисления по заработной плате определяются по следующей формуле:

$$C_{\text{соц}} = K_{\text{соц}} \cdot C_{\text{осн}}$$

где  $K_{\text{соц}}$  – коэффициент, учитывающий размер отчислений из заработной платы. Данный коэффициент составляет 30,2% от затрат на заработную плату и включает в себя:

- отчисления в пенсионный фонд;
- на социальное страхование;

- на медицинское страхование.

Итак, отчисления из заработной платы составили:

$$C_{\text{соц}} = 0,32 \cdot 661826 = 211784,32 \text{ (руб.)}$$

Затраты на электроэнергию включает в себя затраты на электроэнергию при работе оборудования а также затраты на электроэнергию, потраченную на освещение. Затраты на электроэнергию при работе оборудования для технологических целей рассчитываются по формуле:

$$\mathcal{E}_{\text{об}} = P_{\text{об}} \cdot C_{\mathcal{E}} \cdot t_{\text{об}}$$

где  $\mathcal{E}_{\text{об}}$  – затраты на электроэнергию, потребляемую оборудованием, руб.;

$P_{\text{об}}$  – мощность, потребляемая оборудованием, кВт;

$C_{\mathcal{E}}$  – тарифная цена за 1 кВт·час,  $C_{\mathcal{E}} = 2,17$  руб/кВт·час;

$t_{\text{об}}$  – время работы оборудования, час.

Время работы оборудования вычисляется на основе данных для таблицы 5 для инженера из расчета, что продолжительность рабочего дня равна 8 часов.

Мощность, потребляемая оборудованием, определяется по формуле:

$$P_{\text{об}} = P_{\text{уст.об}} \cdot K_{\text{с}}$$

где  $P_{\text{уст.об}}$  – установленная мощность оборудования, кВт;

$K_{\text{с}}$  –коэффициент спроса, зависящий от количества, загрузки групп электроприемников.

Для технологического оборудования малой мощности  $K_{\text{с}} = 1$ .

Затраты на электроэнергию для технологических целей приведены в таблице 6.

Таблица 6 – Затраты на электроэнергию для технологических целей

| Наименование оборудования | Время работы оборудования $t_{\text{об}}$ , час | Потребляемая мощность $P_{\text{об}}$ , кВт | Затраты ЭОБ, руб. |
|---------------------------|-------------------------------------------------|---------------------------------------------|-------------------|
| Персональный компьютер    | 3240                                            | 0,3                                         | 2109,24           |

|        |         |
|--------|---------|
| Итого: | 2109,24 |
|--------|---------|

В статье «Амортизационные отчисления» от используемого оборудования рассчитывается амортизация за время выполнения работы для оборудования, которое имеется в наличии.

Амортизационные отчисления рассчитываются на время использования ЭВМ по формуле:

$$C_{AM} = \frac{H_A \cdot C_{OB}}{F_D} \cdot t_{BT} \cdot n,$$

где  $H_A$  – годовая норма амортизации,  $H_A = 25\%$ ;

$C_{OB}$  – цена оборудования,  $C_{OB} = 31000$  руб.;

$F_D$  – действительный годовой фонд рабочего времени,  $F_D = 5080$  часов;

$t_{BT}$  – время работы вычислительной техники при создании программного продукта,  $t_{BT} = 1448$  часа;

$n$  – число задействованных ПЭВМ,  $n = 1$ .

Итак, затраты на амортизационные отчисления составили:

$$C_{AM} = \frac{0,25 \cdot 31000}{5080} \cdot 1448 \cdot 1 = 2209,06 \text{ (руб)}$$

В статье «Прочие расходы» отражены расходы на разработку проекта, которые не учтены в предыдущих статьях. Прочие расходы составляют 16 % от единовременных затрат на выполнение технического продукта и проводятся по формуле:

$$C_{проч} = (C_{мат} + C_{осн} + C_{соц} + Э + C_{AM}) \cdot 0,16$$

$$C_{проч} = (31000 + 661826 + 211784,32 + 2109,24 + 2209,06) \cdot 0,16 \\ = 145075,13 \text{ (руб)}$$

Проведя расчет сметы затрат на разработку, можно определить общую стоимость разработки НТИ.

Таблица 7 – Смета затрат на разработку НТИ

| Статья затрат                | Условное обозначение | Сумма, руб. |
|------------------------------|----------------------|-------------|
| Материалы и покупные изделия | $C_{мат}$            | 31000       |



|                               |                   |            |
|-------------------------------|-------------------|------------|
| Основная заработная плата     | $C_{\text{осн}}$  | 661826     |
| Отчисления в социальные фонды | $C_{\text{соц}}$  | 211784,32  |
| Расходы на электроэнергию     | $\Xi$             | 2109,24    |
| Амортизационные отчисления    | $C_{\text{ам}}$   | 2209,06    |
| Прочие расходы                | $C_{\text{проч}}$ | 145075,13  |
| <b>Итого:</b>                 |                   | 1054003,75 |

Таким образом, расходы на разработку составили  $C = 1054003,75$  руб.

Прибыль не представляется посчитать возможным, так как нельзя посчитать прямой экономический эффект от внедрения данного проекта .

#### 4.4. Оценка научно-технической результативности НИР

Научно-технический эффект определяет, возможно ли будет использовать результаты работы в других научно-исследовательских и опытно-конструкторских работах, при создании новой продукции. Научно-техническая результативность НИР оценивается путем использования метода балльных оценок. Метод заключается в определении индекса научно-технического уровня на основе оценок признаков работы по следующей формуле [13]:

$$K_{\text{нту}} = \sum_{i=1}^3 R_i \cdot n_i,$$

где  $R_i$  – весовой коэффициент  $i$ -го признака;

$n_i$  – значение количественной оценки  $i$ -го признака в баллах.

Описание факторов, а также значения весовых коэффициентов и оценок в баллах приведены в таблице .

Таблица 8 – Характеристики факторов

| Фактор научно-технического | $R_i$ | Характеристика фактора | $n_i$ |
|----------------------------|-------|------------------------|-------|
|----------------------------|-------|------------------------|-------|

| эффекта                |     |                                                                                                                          |     |
|------------------------|-----|--------------------------------------------------------------------------------------------------------------------------|-----|
| Уровень новизны        | 0,4 | Принципиально новая. Новое направление в науке и технике, новые факты, закономерности, теории и способы                  | 10  |
|                        |     | Новая. Объяснение тех же фактов по-новому, ввод новых понятий, дополняющих полученные ранее результаты                   | 7   |
|                        |     | Относительно новая. Обобщение и систематизация уже имеющихся сведений, выявление новых связей между известными факторами | 4   |
|                        |     | Не обладает новизной. Результат был известен ранее                                                                       | 0   |
| Теоретический уровень  | 0,1 | Установление закона, разработка новой теории                                                                             | 10  |
|                        |     | Глубокий анализ проблемы, анализ взаимодействия между факторами с наличием объяснений                                    | 8   |
|                        |     | Разработка способа (алгоритма, программы и т.д.)                                                                         | 6   |
|                        |     | Элементарный анализ связей между факторами (наличие гипотез, объяснение версий, практические рекомендации)               | 2   |
|                        |     | Предоставление отдельных элементарных факторов, описание наблюдений, опытов, результатов измерений                       | 0,5 |
| Возможность реализации | 0,5 | В течение первых лет                                                                                                     | 10  |
|                        |     | От 5 до 10 лет                                                                                                           | 4   |
|                        |     | Свыше 10 лет                                                                                                             | 2   |

Соответствие качественного уровня НТИ значениям показателей приведены в таблице .

Таблица 9 – Оценки уровней научно-технического эффекта

| Уровень научно-технического эффекта | Значение индекса |
|-------------------------------------|------------------|
| Низкий                              | 1-4              |
| Средний                             | 4-7              |
| Высокий                             | 8-10             |

По данным таблицы индекс научно-технического эффект разработки составит:

$$K_{\text{нту}} = 0,4 \cdot 4 + 0,5 \cdot 6 + 0,1 \cdot 10 = 5,6$$

Согласно таблице , данной величине индекса соответствует средний уровень научно-технического эффекта.

#### **4.5. Выводы по главе 4**

В данной главе определены основные этапы работ над проектом, а также их трудоемкость, построена диаграмма Ганта, визуализирующая ход разработки, составлена смета затрат проекта, проведен SWOT анализ, выполнена оценка научно-технической результативности.

SWOT-анализ дал возможность рассмотреть основные положительные и отрицательные факторы, оказывающие влияние на проект. На его основе предложены возможные способы коррекции слабых сторон проекта и борьбы с внешними угрозами. Полученная величина индекса НТЭ дает возможность назвать проект целесообразным с точки зрения научно-технологической результативности.

По результатам, полученным в ходе выполнения раздела, можно говорить о том, что разработка данного программного обеспечения имеет определенный научно-технический потенциал, позволяющий рассматривать данный проект как привлекательный, с точки зрения коммерческой ценности.

**Министерство образования и науки Российской Федерации**  
**Федеральное государственное бюджетное образовательное учреждение**  
**высшего профессионального образования**  
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**  
**ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

|                                           |                                                                                              |
|-------------------------------------------|----------------------------------------------------------------------------------------------|
| Институт                                  | <b>ЭНИН</b>                                                                                  |
| Направление подготовки<br>(специальность) | <b>Автоматизированные системы диспетчерского<br/>управления электроэнергетических систем</b> |
| Уровень образования                       | <b>Магистратура</b>                                                                          |
| Кафедра                                   | <b>ЭЭС</b>                                                                                   |
| Период выполнения                         | (осенний / весенний семестр 2016/2017 учебного года)                                         |

Студенту:

|               |                         |
|---------------|-------------------------|
| <b>Группа</b> | <b>ФИО</b>              |
| 5AM5Д         | Руденко Николай Юрьевич |

Тема работы:

|                                                                       |  |
|-----------------------------------------------------------------------|--|
| Создание программы дорасчета параметров ТИ и ТС (Калькулятор ТИ, ТС)  |  |
| Утверждена приказом проректора-директора<br>(директора) (дата, номер) |  |

Форма представления работы:

|                                                                                                             |
|-------------------------------------------------------------------------------------------------------------|
| <b>Магистерская диссертация</b><br>(бакалаврская работа, дипломный проект/работа, магистерская диссертация) |
|-------------------------------------------------------------------------------------------------------------|

### ЗАДАНИЕ

|                                                                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|--------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Перечень вопросов, подлежащих исследованию, проектированию и разработке</b> | <p>1. Анализ необходимости тестирования создаваемого программного обеспечения, в том числе оперативно-информационного комплекса СК-2007 и программы дорасчета Калькулятор телеизмерений и телесигналов.</p> <p>2. Анализ причин, вызывающих некорректную работу программного обеспечения. Разбор факторов приводящих к появлению таких причин.</p> <p>3. Анализ причин, приводящих к неправильному функционированию ОИК СК-2007. Возможные пути решения причин некорректной работы программного обеспечения.</p> <p>4. Анализ причин, приводящих к неправильному функционированию программного модуля Калькулятор телеизмерений и телесигналов. Возможные пути решения причин некорректной работы программного обеспечения.</p> |
| <b>Перечень расчётного и графического материала</b>                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>Дата выдачи задания для раздела по линейному графику</b>                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

**Задание выдал консультант:**

|                  |                        |                               |                |             |
|------------------|------------------------|-------------------------------|----------------|-------------|
| <b>Должность</b> | <b>ФИО</b>             | <b>Ученая степень, звание</b> | <b>Подпись</b> | <b>Дата</b> |
| Ассистент        | Невский Егор Сергеевич | -                             |                |             |

**Задание принял к исполнению студент:**

|               |                         |                |             |
|---------------|-------------------------|----------------|-------------|
| <b>Группа</b> | <b>ФИО</b>              | <b>Подпись</b> | <b>Дата</b> |
| 5AM5Д         | Руденко Николай Юрьевич |                |             |

## **Глава 5. Социальная ответственность**

В настоящее время компьютерная техника широко применяется во всех сферах человеческой деятельности, в том числе и в электроэнергетике.

Однако для выполнения компьютером определенных поставленных перед ним задач необходимо, чтобы на нем было установлено специализированное программное обеспечение.

Сами программы являются ничем иным, как подробными и последовательными наборами команд, которые выполняют вычислительные машины. Современные программы включают в себя тысячи строк кода, написанные на одном из языков программирования. От правильности написанного кода зависит и качество выполнения программами своих функций. Качество программного продукта – это степень, в которой система, компонент или процесс удовлетворяют потребностям или ожиданиям заказчика или пользователя. Повышение качества работы программы возможно при тщательном анализе всех возможных проблем, возникающих при работе программы.

Важным фактором надежности программы является время, необходимое для восстановления работоспособности программы после появившейся ошибки. Восстановление может заключаться в исправлении программного кода, исправлении исходных данных или изменении логики вычислительного процесса. Этот фактор можно охарактеризовать одним словом – восстанавливаемость. Он зависит от многих других факторов, таких как сложность структуры программы, алгоритмического языка, на котором производилась разработка программы, стиля программирования и т.д.

Также можно говорить об устойчивости программы, понимая по этим ее способность ограничить последствия сбоя работы из-за собственной ошибки или воздействия внешних факторов (неисправность аппаратной части, поступление некорректных данных, ошибки пользователя и т.д.) и противостоять этим воздействиям.

Устойчивость можно повысить при помощи различных форм избыточности, позволяющих дублирование функций программы, использовании альтернативных программы для решения схожих задач и т.п.

### **5.1. Причины, вызывающие некорректную работу программного обеспечения**

Основными причинами, вызывающими нарушения нормального функционирования программного обеспечения, являются:

- ошибки в программном коде;
- поступление недостоверной входной информации;
- ошибочные действия пользователя;
- неисправности аппаратной составляющей, на которой происходит реализация программного продукта.

Ошибки в программном коде. Специфика создания сложных программных средств состоит в том, что из-за большого объема программного кода, включающего в себя порой десятки тысяч программного кода практически невозможно обнаружить и ликвидировать все ошибки. В результате в программе почти всегда остается некоторое количество ошибок. Они могут вызвать неправильное функционирование программы при определенных условиях. Наличие в программе скрытых ошибок является основным фактором нарушения ее нормальной работы.

Можно выделить следующие основные классы ошибок в программах:

- Ошибки вычислений. Эти ошибки происходят в случае неверного описания выражений. Примером подобной ошибки является неверное преобразование типов переменных или знаков операций.
- Логические ошибки в написании программного обеспечения могут являться причинами различий аналитического и запрограммированного алгоритмов. К ним могут относиться недостаточно детальная проработка кода, отвечающего за связи между различными

компонентами ПО, неверное определение мест вхождения в цикл и выхода из него, неправильное задание условий и многие другие;

- Ошибки ввода-вывода, которые могут являться причинами неверного ввода исходных данных и вывода полученных результатов. К ним относятся неверный выбор числа входных и выходных параметров, а также некорректные алгоритмы обработки информации на входе и выходе программного продукта;

- Ошибки манипулирования данными. Их возникновение обусловлено неправильным заданием первоначальных значений данным, оперирование большим количеством переменных, выполняющих схожие функции, что приводит к запутанности кода и повышает риск использования в операциях лишних данных. Кроме того причиной возникновения ошибок может быть неверное создание переменных с похожими именами, что может привести к затруднению в их использовании;

- Ошибки совместимости связаны с отсутствием совместимости разрабатываемого или применяемого ПО с операционной системой или другими прикладными программами. Как правило данные ошибки возникают при создании программы в одной операционной системе и её использовании в другой без какой-либо переработки кода под выбранную систему.

- Ошибки сопряжений. При создании программного продукта, для работы которого необходим обмен информацией с другими программами, могут возникать ошибки связанные с некорректными алгоритмами отправки или запросов данных сторонних программ.

- Другой причиной, приводящей к нарушению нормальной работы программы, является искажение входной информации. Это возникает в том случае, когда входные данные не соответствуют области допустимых значений для используемых в программе значений переменных.

Причинами подобного нарушения работы могут быть:

- а) нарушение целостности информации на носителе;
- б) помехи в каналах связи при передаче информации;



- с) отказ оборудования приема-передачи информации;
- д) потери информации при буферизации данных;
- е) искажение первичной информации непосредственно пользователем при редактировании.

- Неисправность аппаратуры. Появление отказов или сбоев в работе аппаратуры приводят к нарушению хода обработки информации и, как следствие, могут исказить исходные данные и саму программу.

Наиболее часто встречающимися последствиями появления ошибок в работе программы являются:

- а) аварийное завершение работы программы;
- б) значительное увеличение времени работы программы;
- с) заикливание программы на выполнении некоторой последовательности команд;
- д) нарушение очередности вызова отдельных программ, приводящее к пропуску необходимых программ;
- е) искажение входных, выходных или промежуточных элементов данных

## **5.2. Причины, приводящие к неправильному функционированию ОИК СК-2007**

Оперативно-информационный комплекс СК-2007 представляет собой объединение аппаратной части и программного обеспечения, содержащего в себе большое количество подпрограмм, выполняющих самые разнообразные функции для нужд энергетики. В АО «СО ЕЭС» для поддержания стабильной работы, настройки комплекса, а также устранения возникающих аварийных и нештатных ситуаций задействованы несколько служб. Учитывая масштабность охватываемых комплексом задач, а также присутствующий при разработке человеческий фактор, можно сделать вывод, что СК-2007 содержит в себе некоторое количество ошибок, перечисленных в предыдущем пункте.

Для оценки качества работы оперативно-информационного комплекса и нахождения ошибок, возникающих в процессе эксплуатации, энергетические компании в течение длительного времени проводят многократные испытания каждого компонента СК-2007. Далее компания-производитель данного комплекса, путем поддержания обратной связи с пользователями своего продукта, производит сбор информации о возникающих проблемах в работе СК-2007 и осуществляет их поиск и устранение. Посредством отправки пользователям обновлений программы происходит устранение причин некорректной работы комплекса и повышается качество выполняемых им функций.

Многолетний опыт использования ОИК СК-2007 в СО ЕЭС показывает, что программа практически полностью лишена недостатков в своих алгоритмах по выполнению различного рода задач. Кроме того налажена функция контроля и информирования персонала о некорректном вводе данных. На данный момент обновления, выпускаемые производителем, касаются исправления неточностей в работе вторичных сервисов, незначительно влияющих на работу основных, а также оптимизации работы всего комплекса. Однако при изменении алгоритма работы и влиянии человеческого фактора разработанные новшества могут положительно сказаться на работе одних компонентов и негативно на других. Например, при общем увеличении быстродействия программы, произойдет значительное сокращение сроков хранения информации, что может быть недопустимо для энергетических компаний.

Помимо вышеперечисленных причин отказ работы СК-2007 может произойти по двум факторам, от которых тяжело защитить комплекс. К ним относятся:

- Неверные действия пользователя;
- Неисправность аппаратных средств информационной системы.

Отсутствие внимания или достаточного опыта у эксплуатирующего персонала может привести к нарушению функционирования комплекса, а вся

используемая аппаратура имеет свой рабочий ресурс. Для устранения подобных причин в ОИК СК-2007 предусмотрено резервирование аппаратной части, благодаря чему, поломка какого-либо оборудования не влечет к полному останову технологического процесса, а система авторизации повышает ответственность каждого пользователя при работе с комплексом.

### **5.3. Причины, приводящие к неправильному функционированию программы дорасчета параметров ТИ и ТС**

Программный модуль программы дорасчета параметров ТИ и ТС (Калькулятор ТИ, ТС) предназначен для реализации дорасчетных алгоритмов поступающей из БДРВ ОИК СК-2007 информации, записанных в конфигурационном XML-файле.

Неисправная работа Калькулятора не влечет за собой нарушения функционирования СК-2007, но может повлечь за собой запись не соответствующей действительности информации в БД ОИК. Это может в свою очередь привести к принятию оперативно-диспетчерским персоналом неверных решений.

К возможным причинам нарушения функционирования программы дорасчета параметров ТИ и ТС относятся:

- Ошибочные действия пользователя, другими словами воздействие «человеческого фактора». Примеры: подключение к не тому серверу БД, что повлечет за собой коллизии чтения-записи данных. Чтение и запуск на выполнения ненужного алгоритма дорасчета, также может привести к перезаписи данных на сервере БД и потере нужной информации. Устранение данной проблемы заключается в разработке как можно более простого и интуитивно понятного интерфейса программы, который минимизирует возможные ошибки. Также вывод информации из Калькулятора ТИ и ТС и запись в ОИК происходит с помощью библиотеки

компонента доступа к данным, который создан разработчиком комплекса, а значит, возможность его аномальной работы минимальна

- Неисправность аппаратных средств. Вероятность нарушения работы в данном случае крайне мала, так как для работы программного модуля не требуется специализированного оборудования.

- Ошибки, скрытые в самой программе являются наиболее значительной причиной неправильной работы программы. Устранение этой проблемы ведется с помощью тщательной проработки функционала программы, определения всего объема и типа данных, затрагиваемых в ПО, а также детальной проработки алгоритмов работы программы.

#### **5.4. Выводы по главе 5**

В данной главе произведен общий анализ основных причин и последствий нарушения нормального функционирования программного обеспечения. Рассмотрены причины нарушения функционирования ОИК СК-2007 и программы дорасчета параметров ТИ и ТС, возможные последствия, а также способы их устранения.

Наиболее тяжелые последствия вызывают ошибки вследствие неверных действий пользователей. Такие ошибки наиболее тяжело предусмотреть и исправить, по причине того что они вызывают нарушение целостности данных. В качестве контрмеры предлагается разработка интуитивно понятного интерфейса программы и повышение квалификации пользователей.

## Список используемых источников

1. «АО СО ЕЭС» [Электронный ресурс] URL:[http://so-ups.ru/index.php?id=press\\_release\\_view&no\\_cache=1&tx\\_ttnews%5Btt\\_news%5D=374](http://so-ups.ru/index.php?id=press_release_view&no_cache=1&tx_ttnews%5Btt_news%5D=374), свободный – Яз. рус. Дата обращения: 15.05.2017.
2. Руководство пользователя ОИК СК-2007. Том 6 «Руководство по настройке обработки и хранения данных», 2015 – 103 с.
3. Методические указания по выполнению практических работ по дисциплине «Программные продукты в электроэнергетике». НГТУ им. Р. Е. Алексеева. – Нижний Новгород, 2014 г. - 81 с.
4. Дин Леффингуэлл, Дон Уидриг. «Принципы работы с требованиями к программному обеспечению. Унифицированный подход»: Пер. с англ. – М.: Издательский дом «Вильямс», 2002 г. – 448 с.
5. «C Sharp» [Электронный ресурс] URL:[https://ru.wikipedia.org/wiki/C\\_Sharp](https://ru.wikipedia.org/wiki/C_Sharp), свободный – Яз. рус. Дата обращения: 19.05.2017
6. «Терминология языка C#» [Электронный ресурс] URL:[https://msdn.microsoft.com/ru-ru/library/ms173231\(v=vs.90\).aspx](https://msdn.microsoft.com/ru-ru/library/ms173231(v=vs.90).aspx), свободный - Яз. рус. Дата обращения: 21.05.2017
7. «Понятие объектно-ориентированного программирования (ООП). Классы и объекты» [Электронный ресурс] URL: [http://mycsharp.ru/post/23/2013\\_06\\_20\\_ponyatie\\_obektno-orientirovannogo\\_programmirovaniya\\_\(oop\)\\_klassy\\_i\\_obekty.html](http://mycsharp.ru/post/23/2013_06_20_ponyatie_obektno-orientirovannogo_programmirovaniya_(oop)_klassy_i_obekty.html), свободный - Яз. рус. Дата обращения: 11.05.2017
8. ГОСТ Р ИСО/МЭК 25010-2015 Информационные технологии (ИТ). Системная и программная инженерия. Требования и оценка качества систем и программного обеспечения (SQuaRE). Модели качества систем и программных продуктов Виссер – разработка обслуживаемых программ. введ. 01.06.2016. – М.: Стандартиформ, 2015. – 44 с.

9. Джуст Виссер. Разработка обслуживаемых программ на языке C# / пер. с англ. Р. Н. Рагимова. — М.: ДМК Пресс, 2017. — 192 с.: ил.
10. С.В. Одиночкина. «Основы технологий XML» - СПб: НИУ ИТМО, 2013. – 56 с.
11. Руководство пользователя ОИК СК-2007. Том 9 «Расширение функциональности» , 2015 – 138 с.
12. Гаврикова Н.А. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение: учебно-методическое пособие / Томский политехнический университет. – Томск: Изд-во Томского политехнического университета, 2014. – 73 с.
13. Гольдштейн Г.Я. «Стратегический инновационный менеджмент: Учебное пособие». Таганрог: Изд-во ТРТУ, 2004. – 267 с.